

CLAIMS

1. A method comprising:
identifying one or more characteristics of one or more elements of a media processing system; and
dynamically negotiating which element of a media processing system will perform certain media processing tasks based, at least in part, on the identified one or more characteristics of the system element(s).
2. A method according to claim 1, wherein the method is implemented by an application program interface (API) of an operating system executing on a general purpose computing system hosting the media processing system.
3. A method according to claim 2, wherein identifying the one or more characteristics of the system elements comprises:
generating a negotiation data structure in accordance with a proposed set of media processing capabilities; ;
issuing the negotiation data structure to one or more elements of a media processing system;
determining whether each of the one or more media processing system elements accepted the negotiation data structure; and
executing media processing tasks within the proposed media processing capabilities if the negotiation data structure is accepted by the elements of the media processing system.

1 **4.** A method according to claim 3, further comprising:
2 generating another negotiation data structure in accordance with another
3 proposed set of media processing capabilities , if the media processing system
4 elements did not previously accept the negotiation data structure; and
5 iteratively performing the issuing, determining and generating steps until
6 the media processing system elements accept a negotiation data structure.

7
8 **5.** A method according to claim 3, further comprising:
9 generating operational data structure(s) to pass between the one or more
10 media processing system elements to facilitate shared decoding of multimedia
11 content between media processing system elements.

12
13 **6.** A method according to claim 5, wherein the data structures include at
14 least, one or more residual difference data structures and one or more macroblock
15 control command data structures which have fixed sizes, determined on a per-
16 frame basis.

17
18 **7.** A method according to claim 5, wherein the operational data
19 structures comprise a raw bitstream data structure, dynamically generated to pass
20 media content from a decoder application to a hardware accelerator to facilitate
21 decoding by the accelerator.

22
23 **8.** A method according to claim 3, wherein the data structure(s) include
24 one or more auto-negotiation data structure(s).
25

1 **9.** A method according to claim 8, wherein the auto-negotiation data
2 structure(s) include a connection mode data structure denoting a proposed set of
3 required processing system capabilities that indicate capabilities needed for
4 decoding data in a format specified by a particular media processing standard.
5

6 **10.** A method according to claim 8, wherein the auto-negotiation data
7 structure(s) include a connection configuration data structure denoting a proposed
8 split in media processing between media processing system elements.
9

10 **11.** A method according to claim 1, wherein identifying one or more
11 characteristics of media processing system elements comprises:

12 generating one or more auto-negotiation data structure(s) denoting a
13 proposed set of media processing system capabilities and/or a proposed split in
14 media processing among media system elements; and

15 iteratively issuing the auto-negotiation data structure(s) to media processing
16 system elements until a proposed set of media processing system capabilities
17 and/or split in media processing is agreed upon by each element of the media
18 processing system.
19

20 **12.** A storage medium comprising a plurality of executable instructions
21 which, when executed, implement a method according to claim 1.
22

23 **13.** A computing system comprising:

24 a storage medium having stored therein a plurality of executable
25 instructions; and

1 an execution unit, coupled to the storage medium, to execute at least a
2 subset of the plurality of executable instructions to implement a method according
3 to claim 1.

4
5 **14.** A computing system comprising:
6 a media processing application;
7 a media processing accelerator; and
8 an operating system, executing on the computing system, including an
9 application program interface (API) to facilitate communication between the
10 media processing application and the media processing accelerator, wherein the
11 API includes auto-negotiation data structure(s) and operational data structure(s) to
12 dynamically negotiate at least a set of media processing system capabilities and/or
13 a split in media processing among system elements suitable to each of the media
14 processing application and the media processing accelerator and to processing of
15 received media content, respectively.

16
17 **15.** A computing system according to claim 14, the auto-negotiation
18 data structures comprising:

19 a processing mode data structure, generated by the API to propose a media
20 set of media processing capabilities and/or a split in media processing among
21 system elements to each of the media processing application and the media
22 processing accelerator.

1 **16.** A computing system according to claim 15, wherein the processing
2 mode data structure is a ConnectMode data structure.

3
4 **17.** A computing system according to claim 14, the auto-negotiation
5 data structures comprising:

6 a connection mode data structure, specifying a set of media processing
7 system capabilities, and/or

8 a processing configuration data structure, generated by the API to propose a
9 split in media processing between the media processing application and the media
10 processing accelerator.

11
12 **18.** A computing system according to claim 17, wherein the processing
13 configuration data structure comprises a ConnectConfig data structure.

14
15 **19.** A computing system according to claim 14, the operational data
16 structure(s) comprising:

17 one or more residual difference data structures, generated by the API to
18 pass residual difference information between the media processing application and
19 the media processing accelerator for media processing; and

20 one or more control command data structures, generated by the API to pass
21 control commands between the media processing application and the media
22 processing accelerator.

1 **20.** A computing system according to claim 19, wherein the residual
2 difference data structures and the control command data structures are
3 dynamically generated when the auto-negotiation data structures have negotiated a
4 split in media processing between the media processing application and the media
5 processing accelerator.

6
7 **21.** A computing system according to claim 14, the operational data
8 structure(s) comprising:

9 a raw bitstream data structure, generated by the API to pass media content
10 in raw bitstream form from the media processing application to the media
11 processing accelerator.

12
13 **22.** A computing system according to claim 21, wherein the raw
14 bitstream data structure is dynamically generated when the auto-negotiation data
15 structures have negotiated that the media processing accelerator will perform the
16 media processing.

17
18 **23.** A computing system according to claim 14, further comprising:
19 a storage medium including a plurality of executable instructions; and
20 an execution unit, coupled to the storage medium, to execute at least a
21 subset of the plurality of executable instructions to implement the operating
22 system and associated API.

1 **24.** A computing system according to claim 23, wherein the execution
2 unit executes at least a subset of the plurality of executable instructions to
3 implement the media processing application.

4
5 **25.** A storage medium comprising a plurality of executable instructions
6 including instructions which, when executed, implement an application program
7 interface (API) to facilitate media processing between media processing system
8 elements, the API including one or more auto-negotiation data structure(s)
9 dynamically generated and issued to the media processing system elements to
10 negotiate a set of media processing capabilities and/or a split in media processing
11 tasks between the elements of the media processing system suitable to each of the
12 media processing system elements.

13
14 **26.** A storage medium according to claim 25, wherein the auto-
15 negotiation data structures include a processing mode data structure, iteratively
16 issued to the media processing system elements denoting an iteratively changing
17 the proposed set of media processing capabilities until accepted by all media
18 processing system elements.

19
20 **27.** A storage medium according to claim 25, wherein the auto-
21 negotiation data structures include a processing configuration data structure,
22 iteratively issued to the media processing system elements denoting an iteratively
23 changing proposed split in media processing between media processing system
24 elements.

1 **28.** A storage medium according to claim 25, wherein the API further
2 comprises:

3 one or more operational data structures, to facilitate media processing
4 between media processing system elements based, at least in part, on a negotiated
5 set of media processing capabilities and/or split in media processing responsibility
6 between the media processing system elements.